IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Li, Jianghao	Attorney Docket No.: TRNDP00
Application No.: 10/042,804	Examiner: ZIA, Syed
Filed: October 29, 2001	Group: 2131
Title: SCRIPTING VIRUS SCAN ENGINE	Confirmation No.: 5044

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence is being transmitted to the U.S. Patent and Trademark Office, Central Facsimile Telephone number (571) 273-8300 on this day June 29, 2007.

Signed: Ann Lowe

Ann Lowe

RESPONSE TO NOTICE OF NON-COMPLIANT APPEAL BRIEF
(37 CFR 1.121)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This submission is in response to the Notification of Non-compliant Appeal Brief mailed June 1, 2007 (copy attached) in which the response required by 37 CFR 1.121 was deemed incomplete "because the Appeal Brief does not contain a concise explanation of the subject matter defined in each of the independent claims involved in the appeal." It is believed that this submission, "Summary of Claimed Subject Matter" addresses the objection.

Respectfully submitted,

BEYER WEAVER LLP

Jonathan O. Scott

Beyer Weaver LLP
P.O. Box 70250
Oakland, CA 94612-0250
Telephone: (612) 252-3330
Facsimile: (612) 825-6304

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Ex Parte Li

Application for Patent: 10/042,804

Filed: October 29, 2001

Group Art Unit: 2131

Examiner: Zia, Syed

For:

SCRIPTING VIRUS SCAN ENGINE

APPEAL BRIEF – SUMMARY OF CLAIMED SUBJECT MATTER

BEYER WEAVER LLP
P.O. Box 70250
Oakland, CA 94612-0250
Attorneys for Appellant

CERTIFICATE OF FACSIMILE TRANSMISSION
I hereby certify that this correspondence is being transmitted to the U.S. Patent and Trademark Office, Central Facsimile Telephone number (571) 273-8300 on this day June 29, 2007.

Signed: _____
Ann Lowe

5. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claims 5 and 15 of the present application are drawn to the *generation* of a virus signature for a scripting virus and its polymorphs that can reliably detect the virus and its polymorphs. Independent claim 25 is drawn to a computer readable medium containing program code for generating a virus signature for a scripting virus and its polymorphs that can reliably detect the virus and its polymorphs.

Independent claims 1 and 11 of the present application are drawn to the *detection* of a scripting virus and its polymorphs. Independent claim 21 is drawn to a computer readable medium containing program code for detecting a scripting virus and its polymorphs.

Claims 3 and 4 are dependent on claim 1 and further limit claim 1.

Following is a mapping of the each of the independent claims (1, 5, 11, 15, 21 and 25) and each of the dependent claims argued separately (3 and 4) to the specification by page and line number and to the drawings by reference characters.

Claim 1 recites a method for identifying a computer virus in interpreted language source code (204 and Fig. 5A) using a virus scan engine (200). The method (Fig. 4) involves receiving (302) a portion of interpreted language source code (204 and Fig 5A) (page 10, lines 24-26), generating a language-independent representation of the portion of the interpreted language source code (page 11, lines 17-20), comparing (312) the language-independent representation with a virus signature in a pattern matcher (218) (page 16, lines 30-31 though page 17, line 1), and determining (312) if the language-independent representation matches the virus signature, whereby a match indicates a computer virus has been identified (page 17, lines 8-12).

Claim 3 recites the method of claim 1 with the limitation that the virus signature is a language-independent representation of an interpreted language source code computer virus (page 4, line 29 though page 5, lines 1-2).

Claim 4 recites the method of claim 1 with the limitation that the portion of interpreted language source code and the virus signature are represented as a linearized string of key actions (page 14, lines 26-28 and page 15, lines 25-26).

Claim 11 recites a method for identifying a virus in interpreted language source code (204 and Fig. 5A) using a virus scan engine (200). The method (Fig. 4) involves receiving (302) a portion of interpreted language source code (204 and Fig. 5A) (page 10, lines 24-26), parsing (304) the portion of the interpreted language source code (204 and Fig. 5A) into tokens to generate a tokenized source code (204' and Fig. 5B) using a parser (206), wherein at least some of the tokens represent key actions (page 11, lines 1-3 and lines 8-10), extracting (306) selected key actions from the tokenized source code (204' and Fig. 5B) (page 11, lines 17-18), linearizing (306) the key actions to generate an executing thread (210 and Fig. 7) (page 11, lines 18-20), comparing (312) the executing thread (210 and Fig. 7) with a virus signature of a known virus in a pattern matcher (218) (page 16, lines 30-31 though page 17, line 1), and determining (312) whether the executing thread (210 and Fig. 7) matches the virus signature (page 17, lines 8-12).

Claim 21 describes a computer readable medium containing program code for identifying a computer virus in interpreted language source code (204 and Fig. 5A) using a virus scan engine. The program code comprises instructions (Fig. 4) for receiving (302) a portion of interpreted language source code (204 and Fig. 5A) (page 10, lines 24-26), parsing (304) the portion of the interpreted language source code (204 and Fig. 5A) into tokens to generate a tokenized source code (204' and Fig. 5B) using a parser (206), wherein at least some of the tokens represent key actions (page 11, lines 1-3 and lines 8-10), linearizing (306) at least a portion of the key actions to generate an executing thread (210 and Fig. 7) (page 11, lines 18-20), comparing (312) the executing thread (210 and

Fig. 7) with a virus signature of a known computer virus in a pattern matcher (218) (page 16, lines 30-31 though page 17, line 1), and determining (312) whether the executing thread matches the virus signature (page 17, lines 8-12).

Claim 5 recites a method for generating a virus signature using a virus scan engine (200). The method (Fig. 4) involves receiving (302) a portion of interpreted language source code containing a computer virus (204 and Fig. 5A) (page 14, lines 16-18), parsing (304) the portion of interpreted language source code (204 and Fig. 5A) into tokens using a parser (206) (page 14, lines 19-20), inputting (306) at least a portion of the tokens into a threadizor (208) (page 14, line 23), generating (306) a language-independent representation of the computer virus from the portion of the tokens using the threadizor (208), wherein the language independent representation is a linearized string of key actions (210 and Fig. 7) (page 14, lines 23-25), and storing (310) the language-independent representation of the computer virus as a virus signature (page 16, lines 7-8).

Claim 15 recites a method for generating a virus signature from a portion of interpreted language source code including a computer virus (204 and Fig. 5A) using a virus scan engine (200). The method (Fig. 4) involves receiving (302) a portion of interpreted language source code containing a computer virus (204 and Fig. 5A) (page 14, lines 16-18), parsing (304) the portion of the interpreted language source code containing the computer virus (204 and Fig. 5A) into tokens to generate tokenized source code (204' and Fig. 5B) using a parser (206), wherein at least some of the tokens represent key actions (page 14, lines 19-20), extracting key actions from the tokenized source code (204' and Fig. 5B) (page 15, lines 12-17), linearizing (306) the key actions to generate an executing thread (210 and Fig. 7) using a threadizor (208) (page 15, lines 18-24), determining the set of minimum key actions in the executing thread (210 and Fig. 7) required to effect the computer virus (page 12, lines 13-15), and storing (310) the set of minimum key actions as a virus signature (page 16, lines 7-8).

Claim 25 describes a computer readable medium containing program code for generating a virus signature from a portion of interpreted language source code including a computer virus (204 and Fig. 5A) using a virus scan engine (200). The program code comprises instructions (Fig. 4) for receiving (302) a portion of interpreted language source code containing a computer virus (204 and Fig. 5A) (page 14, lines 16-18), parsing (304) the portion of the interpreted language source code containing the computer virus (204 and Fig. 5A) into tokens to generate tokenized source code (204' and Fig. 5B) using a parser (206) , wherein at least some of the tokens represent key actions (page 14, lines 19-20), linearizing (306) at least a portion of the key actions to generate an executing thread (210 and Fig. 7) using a threadizor (208) (page 15, lines 18-24), determining the set of minimum key actions in the executing thread (210 and Fig. 7) required to effect the computer virus (page 12, lines 13-15), and storing (310) the set of minimum key actions as a virus signature (page 16, lines 7-8).